

AD-A282 722



**FINAL TECHNICAL REPORT**

**—DRAFT—**

1

**S** DTIC  
ELECTE  
JUL 26 1994  
**F**

REPORT No. FR-0142-94-001

July 12, 1994

---

**GUIDANCE, NAVIGATION, AND CONTROL DIGITAL  
EMULATION TECHNOLOGY LABORATORY**

Contract No. DASG60-89-C-0142

Sponsored By

The United States Army Space and Strategic Defense Command

---

**COMPUTER ENGINEERING RESEARCH LABORATORY**

Georgia Institute of Technology

Atlanta, Georgia 30332-0540

This document has been approved  
for public release and sale; its  
distribution is unlimited.

---

Contract Data Requirements List Item A005

Period Covered: Sept 28, 1989–Sept 27, 1994

Type Report: Final (Draft)

94-23290



178

94 7 25 156

## DISCLAIMER

**DISCLAIMER STATEMENT**—The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

---

## DISTRIBUTION CONTROL

- (1) **DISTRIBUTION STATEMENT**—Approved for public release; distribution is unlimited.
- (2) This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227-7013, October 1988.
- 

Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification .....	
By .....	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

# ***FINAL TECHNICAL REPORT***

***—DRAFT—***

**July 12, 1994**

---

**Cecil O. Alford**

## **COMPUTER ENGINEERING RESEARCH LABORATORY**

**Georgia Institute of Technology**

**Atlanta, Georgia 30332-0540**

---

**Dr. Latika Becka**

**USASSDC**

**Contract Monitor**

**Cecil O. Alford**

**Georgia Tech**

**Project Director**

---

**Copyright 1994**

**Georgia Tech Research Corporation**

**Centennial Research Building**

**Atlanta, Georgia 30332-0540**

# FINAL TECHNICAL REPORT—DRAFT

## 1. Overview

The contract began September 28, 1989 and will terminate September 27, 1994. A no-cost extension has been requested. If this is granted, the termination date will be extended and the final report will be moved to a later date. If not, this draft will serve as a basis for completing the final report.

The contract began with seven tasks; (1) Digital Emulation Facility, (2) FPA Seeker Emulator Development, (3) Special Studies, (4) Software Development, (5) Automated Input, (6) PFP Technology, and (7) GN&C Processor Development. These tasks were developed through the first two years of the contract when virtually all funding was removed. Two Annual reports, consisting of 7 volumes each, report on the technical work during these two years [1,2,3,4,5,6,7,8,9,10,11,12,13,14]. These reports were delivered to USASDC as AR-0142-90-001 and AR-0142-91-002. Each volume in each year, corresponds to one of the tasks. No additional reporting on these tasks will be presented in this report.

Two additional tasks have been developed since the funding cut. The first was a speed test on the rad-hard FPU chip developed by Harris. A summary of this testing and the associated report is given in Section 2. More detail is available in the Special Technical Report, STR-0142-93-015 [15].

The second task is the development of an FPA Test System. This task is in progress and is scheduled to complete on Sept. 27, 1994. However, if the contract is extended, the completion date will be pushed out approximately two months on the first system and another month on the second system. It will be quite difficult to complete these systems on the specified due date. Section 3 gives a status report on the FPA Test System. Additional Special Technical Reports will be available after the hardware is delivered if the contract is extended.

## 2. Rad Hard FPU Chip Testing

Georgia Tech got three hardened chips from Harris: FPU#1, FPU#2 and FPU#3. Only two chips were tested—FPU#1 and FPU#2. The third chip (FPU#3) gave erroneous results at any frequency. This may have been caused by inserting the chip into the socket incorrectly due to an incorrect pin diagram. The second chip (FPU#2) was also inserted into the socket incorrectly. However, it did not stay in that configuration for verylong and the only damage was a disabling of the fourth bit on the chip output. This was remedied by modifying the test software to mask out that bit. A commercial version of the FPU was also tested for comparison purposes. The test results follow.

### 2.1. Rad Hard Chips

Georgia Tech received three rad-hard chips from Harris which were labeled FPU #1, FPU #2, and FPU #3. In the process of building the FPU test board there was an error in the pin-out information sent to Georgia Tech from Harris. Chip lead 214 was designated as a "no connection" on the pin-out. Georgia Tech grounded all the pins which were so marked. When the chip was inserted into the socket the lead on this pin melted. After getting a second pin-out listing from Harry Diamond Laboratories it was found that this lead is tied to Vcc. Appropriate changes were made on the board to correct this problem. This chip, FPU #3, would not give correct results at any frequency and was discarded. FPU #2 had a stuck bit on the FPU output, bit 4. However the chip functioned correctly, so the software test was modified to ignore this bit. The speed tests were run using the same tests without checking this particular output bit. It is possible that the results could be too high for any of the tests on this chip. This could happen if this particular bit was in error, but no other bit in the output was in error. For this case, we would accept the output as correct. FPU #1 was never used in any of the preliminary testing. It was only inserted into the test board after all the debugging had been completed and the other rad-hard chip was running correctly. FPU #1 is somewhat faster than FPU #2 on virtually every test and voltage condition.

Table 1. COMMERCIAL-RAD HARD FPU TEST

TEST	4.5 Volts			5.0 Volts			5.5 Volts		
	Com	#1	#2	Com	#1	#2	Com	#1	#2
Logical	21	22	21	22	21	22	21	21	21
Shift	21	21	20	22	21	21	21	21	21
Integer Add	18	15	12	21	17	14	22	16	16

Integer Multiply	20	14	11	22	15	13	22	17	14
Floating Add	15	12	10	17	13	12	16	15	13
Floating Multiply	18	**	9	18	14	9	16	16	9
Pack Exp & Float	17	16	13	21	18	15	20	20	17
Seed, Unpack, Root	17	14	12	19	16	14	21	16	15
Round & Truncate	15	12	10	18	14	12	20	15	13
Sign Manipulation	24	22	21	24	21	21	21	21	21

\*\* The value was below 10 Mhz and could not be tested with the available clock.

## 2.2. Test Results

The test results are shown in Table 1 for the commercial chip (Com) and two of the rad-hard chips (#1,#2). The commercial chip was designed on the Genesil Silicon Compiler and fabricated by NCR using their 1.25 micron CMOS process. This chip was one of the early designs and did not use some of the primitives that were available for later designs. The design was projected to run at 6.6 Mhz. according to the Genesil timing simulator.

The rad-hard FPU chips were designed using a new version of the Genesil silicon compiler. Harris Corporation, with the help of Silicon Compiler Systems Corp., built a rad-hard version of the Genesil compiler. This compiler implemented some of the primitives in the commercial compiler using the Harris CMOS/SOS 1.25 micron process as the target foundry. The hardened chips, in a non-active mode, use about half the power of the commercial FPU chip. During testing with 5 volts applied to each chip, the hardened FPU used around 100 milliamp. of current compared to 200 milliamp. for the commercial chip. Under test the commercial chip did not change in current consumption, but the rad-hard chip increased to 300 milliamps.

Tests were separated into 10 specific categories. These are indicated as (1) logical, (2) shift, (3) integer addition, (4) integer multiplication, (5) floating point addition, (6) floating point multiplication, (7) pack exponent and float, (8) generate seed, unpack exponent, unpack mantissa, square root exponent, and square root mantissa, (9) round or truncate a result, and (10) sign manipulation. The commercial chip performs faster than the rad-hard version. The two hardened FPU chips produced speed characteristics that differed greatly. This could be due to fabrication quality variations from die to die, or the way the chips were selected for sending to Georgia Tech. To establish more rigid comparisons it will be necessary to quantify several of the commercial chips as well as the rad-hard versions. However, this test does give a sense of the rad-hard performance relative to commercial CMOS performance.

## 3. FPA Test System

### 3.1. Design Overview

The Georgia Tech FPA test system is being developed to analyze the characteristics and quality of an FPA sensor. It also allows a user to study the effectiveness of the FPA sensor when using various signal and image processing functions. The primary features supported by the Georgia Tech system are the ability to:

- (1) interface a wide range of FPAs using a specification defined by USASSDC,
- (2) display the raw FPA image live on a color monitor to enable a user to visually locate bad detectors, and to compare and characterize the quality of an FPA sensor,
- (3) select or program any of the four signal/image filters provided (non-uniformity compensation, temporal filtering, spatial filtering, and thresholding), and
- (4) display the intermediate filtered frame outputs in real time, at a refresh rate that does not strain the eyes (minimum 15 screen updates per second).

The system is designed to support FPA Frame sizes up to 1024 x 1024 with real-time display of any 128 x 128 or 256 x 256 segment. System development involves hardware and software. The hardware consists of four board designs (see Figure NO TAG), GT-SIB (SBus Interface Board), GT-FSPB (FPA Signal Processor Board), GT-FIM (FPA Interface Module), and GT-FFE (FPA Frame Emulator). The software consists of a graphical user interface (GUI), GT-XSPI (X Windows-based Signal Processing Interface), designed to run on a SUN SPARC platform. The SPARCstation 20 Model 50 SX with a 17", 1152x900 resolution, 24-bit color monitor has been chosen as the host platform for the Georgia Tech FPA test system.

The GT-SIB interfaces the SPARCstation 20 SBus port with the GT-FSPB. It plugs into one of the SBus connectors inside the SPARCstation 20 CPU box. A custom connector is provided at the back of the box for connection to an external chassis (via

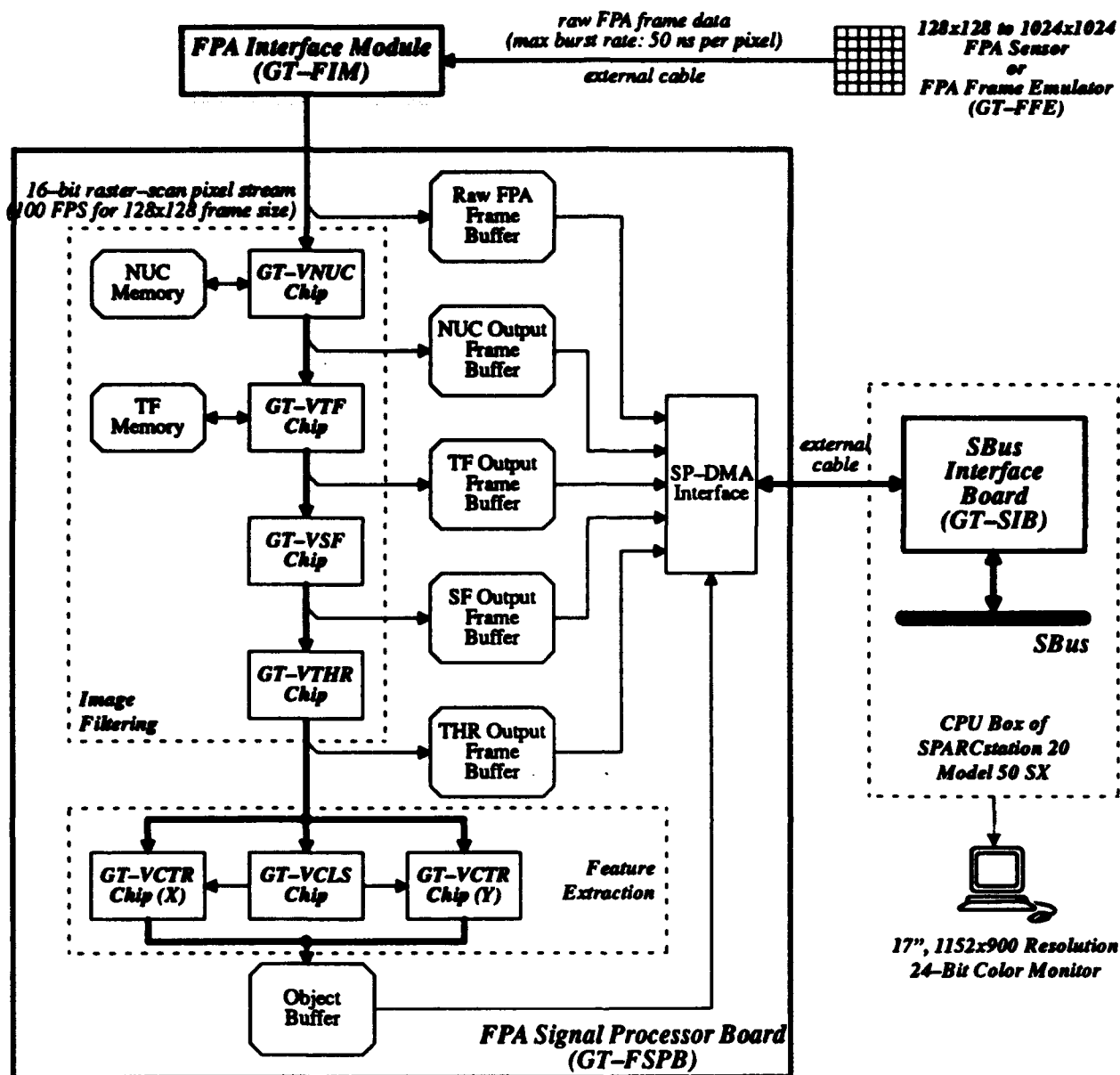


Figure 1. Georgia Tech FPA Test System Hardware

a short cable) containing the GT-FSPB, GT-FIM, and GT-FFE. The external chassis contains its own power supply and fan. The GT-FSPB is the hardware core of the GT FPA Test System, containing the Georgia Tech signal processing (SP) chip set. To accommodate different FPA sensors, the FPA interface circuitry is implemented on a modular, interchangeable daughter board, GT-FIM. An external FPA connector is used to connect to the FPA's data acquisition system. To test and emulate a real FPA sensor, the GT-FFE is used to generate FPA interface signals (Pixel\_Data, Pixel\_Sync, and Frame\_Sync) that exactly meet or exceed the NRaD FPA interface specifications.

The GT-XSPI custom-designed graphical software is based on the XView and MIT X Shared Memory Extension graphics routines. The GT-XSPI allows a user to analyze, in *real time*, the characteristics of an FPA sensor and the effectiveness of each image filtering process with respect to the sensor.

### 3.2. System Development

One of the two SPARCstation 20 workstations to be used as the FPA Test System host has just arrived. The operating system (Solaris 2.3) is currently being installed. The other SPARCstation 20 will be ordered shortly. The following sections describe the status of hardware and software development.

### 3.2.1. Hardware Development

#### 3.2.1.1. GT-SIB (Georgia Tech SBus Interface Board)

The GT-SIB's hardware design and components have been specified. Figure 2 shows the hardware circuit diagram. It contains an SBus DMA Controller chip (LSI Logic L64853A), an 8Kx8 EPROM (Microchip 27HC641) for storing the board's SBus ID, two 8-bit registered transceivers (IDT 74FCT646) for 16-bit data transfers, two 8-bit registers/DFFs (IDT 74FCT574) for the command register, several EPLDs for control/interface logic, and an external connector (around 50 pins) for the connection to GT-FSPB.

The GT-SIB reads in *real time* the object features and pixel intensities of raw/filtered FPA frames from the GT-FSPB, which then transfers this real-time data to the host via the SBus. Continuous data updates are made possible by the "DMA block chaining" feature of the L64853A. The maximum data size of single real-time updates is 184,320 bytes, which consists of 16-bit pixel intensities of five 128x128 frames and 10-byte object features (total object area, total object intensity, X- and Y-coordinate of area centroid, X- and Y-coordinate of intensity centroid) of 4,096 objects. The L64853A performs DMA transfers at a maximum rate of 8.3 Mbytes/sec on a 25-MHz SBus. This bandwidth gives a maximum of 45 real-time data updates per second from the GT-FSPB to the host's display memory. Meeting the real-time display rate at the host's monitor (minimum 15 screen updates per second) should not be difficult.

In addition to reading real-time frame/object data, the host can perform other tasks on the GT-FSPB and GT-FIM by writing the proper command/instruction to the GT-SIB command register. The complete list of commands is shown in Table 1, which includes initialization and diagnostics of the Georgia Tech SP chips, initialization of GT-FIM's processing mode, setting of the Pixel-Address Map, and system reset. Note that all data transfers occur via D16\_Data[15:0] lines.

Table 1. Command Register

<i>Opcode [3:0]</i>	<i>Operand [11:0]</i>	<i>Command Description</i>
0000	-	No operation (idle mode)
0001	Buffer_En[5:0]	Real-time frame/object data read of participating buffers specified in Buffer_En[5:0] = FPA_EN, NUC_EN, TF_EN, SF_EN, THR_EN, CTR_EN
0010	Host_Addr[5:0]	Write to GT-VNUC chip (calibration data, control registers, internal busses, etc.)
0011	Host_Addr[5:0]	Read from GT-VNUC chip (calibration data, internal buses, self-test data, etc.)
0100	Host_Addr[7:0]	Write to GT-VTF chip (IIR filter coefficients, control/test instruction, etc.)
0101	Host_Addr[7:0]	Read from GT-VTF chip (IIR filter coefficients, internal results, self-test data, etc.)
0110	Host_Addr[7:0]	Write to GT-VSF chip (filter coefficients, control/test instruction, etc.)
0111	Host_Addr[7:0]	Read from GT-VSF chip (filter coefficients, internal results, self-test data, etc.)
1000	Host_Addr[3:0]	Write to GT-VTHR chip (thresholding mode, coefficients, etc.)
1001	Host_Addr[3:0]	Read from GT-VTHR chip (control register, coefficients, internal results, etc.)
1010	-	Sequential write to PAM memory
1011	-	Sequential read from PAM memory (used during memory test)
1100	-	Write to GT-FIM control register where D16_Data[9:0] = FIM_Control[9:0] = TDI_sel[2:0] (1/2/4/8/16), Seg_size (128x128/256x256), Seg_Sel[5:0] (0/1/.../63)
1101	-	R E S E R V E D
1110	-	Write synthetic pixel data stream to GT-FIM (connect the external port of a second GT-SIB directly to the FPA port of GT-FIM, D16_Data = Pixel_Data)
1111	-	FPA Test system reset

#### 3.2.1.2. GT-FSPB (Georgia Tech FPA Signal Processing Board)

The GT-FSPB's hardware design and components have been specified. Figure 3 shows the hardware circuit diagram. It consists of the Georgia Tech signal processing (SP) chip set (total 7 ASICs), sixteen 64Kx4 SRAM chips (Cypress CY7C192), eight 16Kx4 SRAM chips (Cypress CY7C162), nine 64Kx16 SRAM modules (Cypress CYM1622), two line drivers (IDT

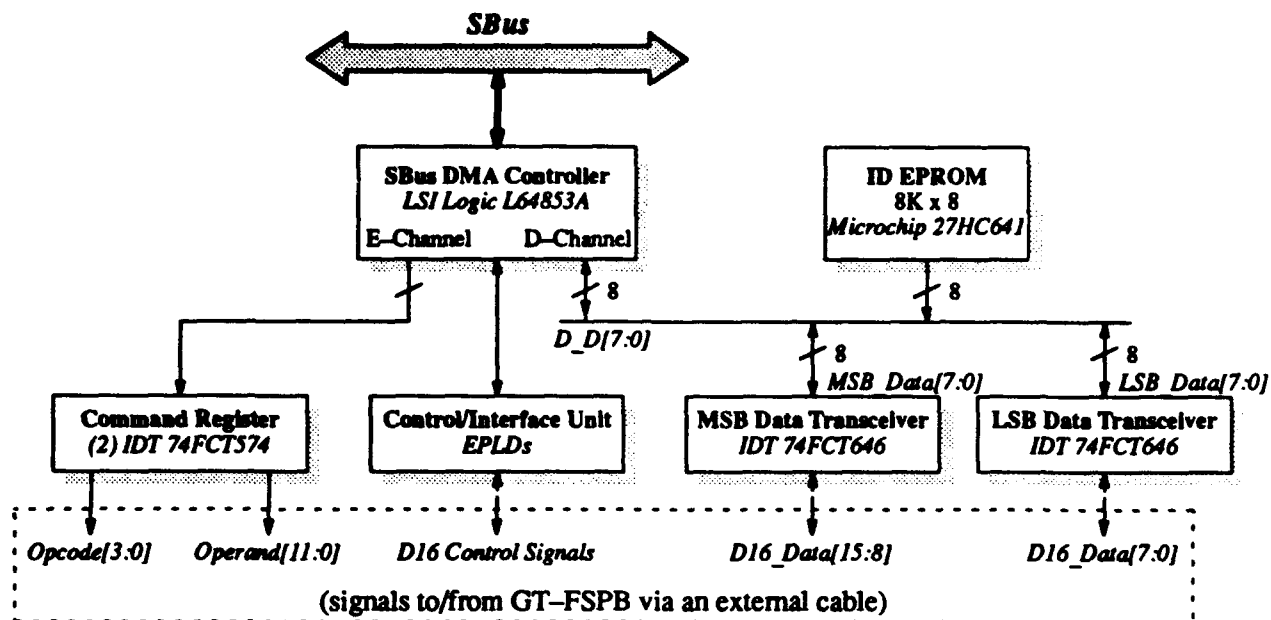


Figure 2. Georgia Tech SBus Interface Board (GT-SIB)

74FCT244) for opcode and operand, two 8-bit registered transceivers (IDT 74FCT646) for 16-bit data transfers, several EPLDs for control logic, an external connector to the GT-SIB, and a daughter-board connector to the GT-FIM. Note that each frame/object buffer is implemented by static RAMs with *separate I/O*. This allows a buffer to be read by the host for real-time display without interrupting the writing of current output frame/object data by the SP chip.

Processing will be done on 128x128 frames at frame rates up to 100 fps. If the FPA has a larger frame size (up to 1024x1024 frame size is supported), processing will be done on a selected part of the FPA (called a *frame segment*) which is 128x128 in size. This will keep the processing rates consistent with the signal processing chip set. Since the NUC and TF chips can handle 128x128 and 256x256 frame sizes, a second processing mode with 256x256 frame segments will be included. For a 256x256 FPA sensor, SF, THR, CLS, and CTR chips will be disabled and only NUC and TF will be in the processing chain. This will provide full 256x256 processing of these frames at a reduced frame rate (the rate will be around 25 Hz). The other SP chips were built with internal memory elements that prohibit them from being used on these larger frame sizes.

### 3.2.1.3. GT-FIM (Georgia Tech FPA Interface Module)

The GT-FIM's hardware design and components have been specified. Figure 4 shows the hardware circuit diagram. It consists of a Xilinx XC4000 FPGA to implement the TDI and control logic circuitry, four 1Mx9 SRAM modules (Cypress CYM1560) for the 1024x1024 input frame buffers, and three 64Kx16 SRAM modules (Cypress CYM1622) for the pixel-address map and 256x256 running sum memories, six 8-bit DFFs/registers for various pixel and control registers, a connector to the GT-FSPB motherboard, and a 40-pin connector for connection to the NRaD FPA data acquisition system. The FPA interface signals (Pixel\_Data, Pixel\_Sync, and Frame\_Sync) are described further in Section 3.3.

The GT-FIM supports several processing modes based on the control register configuration (TDI\_sel[2:0], Seg\_size, and Seg\_sel[5:0]). Five TDI (Time Delay and Integration) factors can be selected: 1, 2, 4, 8, and 16 (number of frames that are time-averaged). Two frame segment sizes are supported, 128 x 128 and 256 x 256. The segment location to be monitored in the overall FPA frame is user programmable. Another useful feature supported by the GT-FIM is the mapping of a random pixel-input-order onto the raster-scan pixel stream. This is done by the Pixel-Address Map (PAM) memory, which is loadable from the host.

During the initial system test when we are not using a real FPA sensor, we will simulate the input FPA image by connecting the GT-FIM's FPA connector to a second GT-SIB. This allows the host to generate the input FPA frames synthetically and send the pixel stream to the Georgia Tech FPA Test System, via the SBus connection at the second GT-SIB.

### 3.2.1.4. GT-FFE (FPA Frame Emulator)

There are two methods for testing the Georgia Tech FPA Test System. The first method is to use a second SBus Interface Board (GT-SIB) to stream in synthetic pixel data on the SBus. The maximum pixel burst rate is 6.67 MHz (150 ns per pixel), which is below the 20 MHz burst rate specification. However, scene data with very large numbers of frames can be generated and stored on the SPARCstation 20 host. The advantage to this method is flexibility. The user can play the scene data through



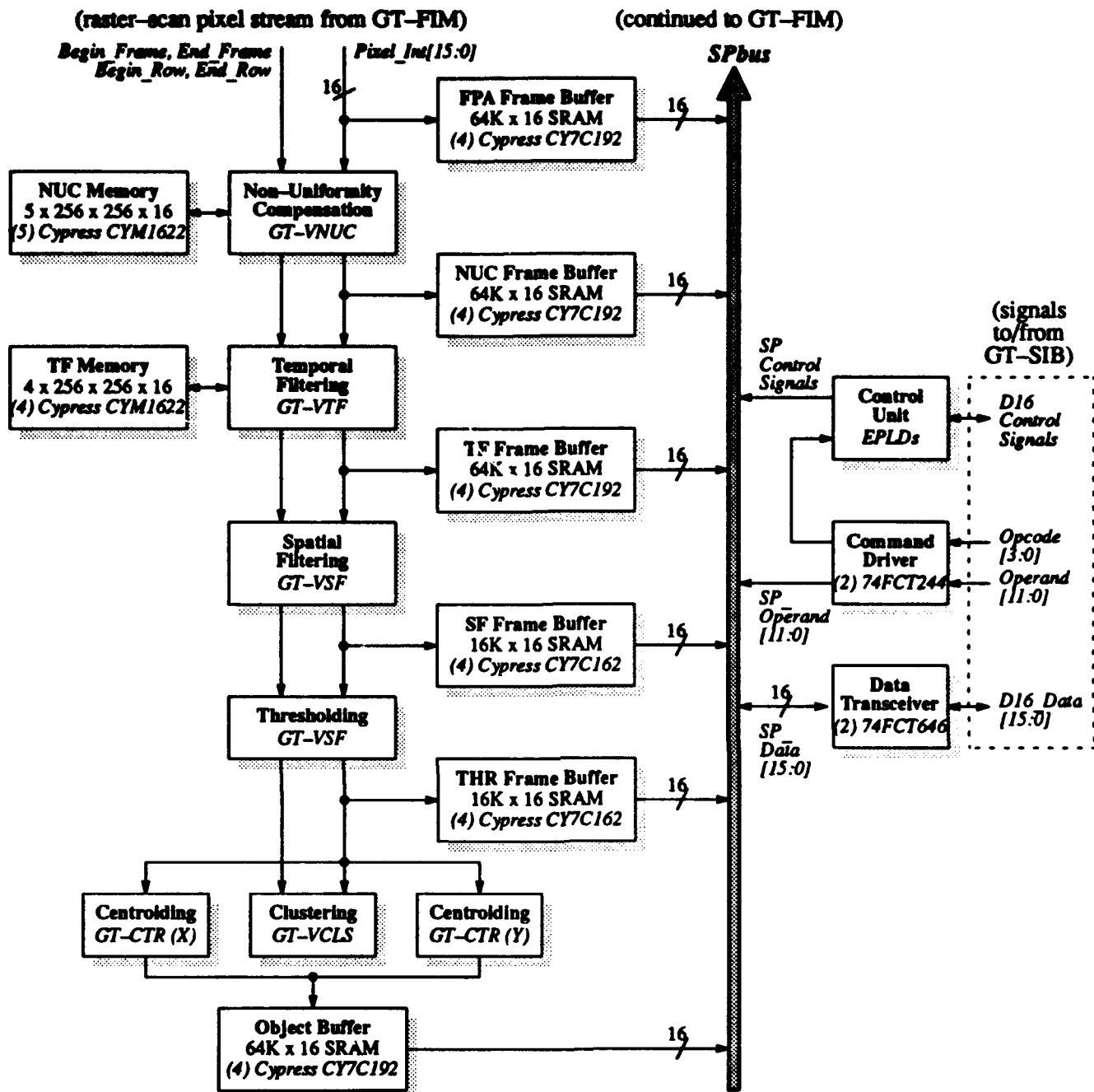


Figure 3. Georgia Tech FPA Signal Processing Board (GT-FSPB)

the Georgia Tech signal processing chip set and investigate the effectiveness of algorithms. The disadvantage is that this method does not test the *real-time* capability of the system.

The second method is to build an FPA Frame Emulator (GT-FFE) board which will be mounted in the external chassis with GT-FSPB and GT-FIM. The GT-FFE will generate all the control signals and pixel data as though it were the FPA sensor system. The signals (*Frame\_Sync*, *Pixel\_Sync*, *Pixel\_Data[15:0]*) will be sent out on a connector through a ribbon cable back into the GT-FIM input connector port. The frame buffer in GT-FFE can store up to 32, 128x128, 16-bit frames of data or eight 256x256 frames. A counter will be used to return to the first frame when the last frame is sent. The pixel burst rate is 20 MHz (one pixel data every 50 ns) with a pause option between rows and frames. This would test the real-time performance of the Georgia Tech FPA Test System using the NRaD input specifications.

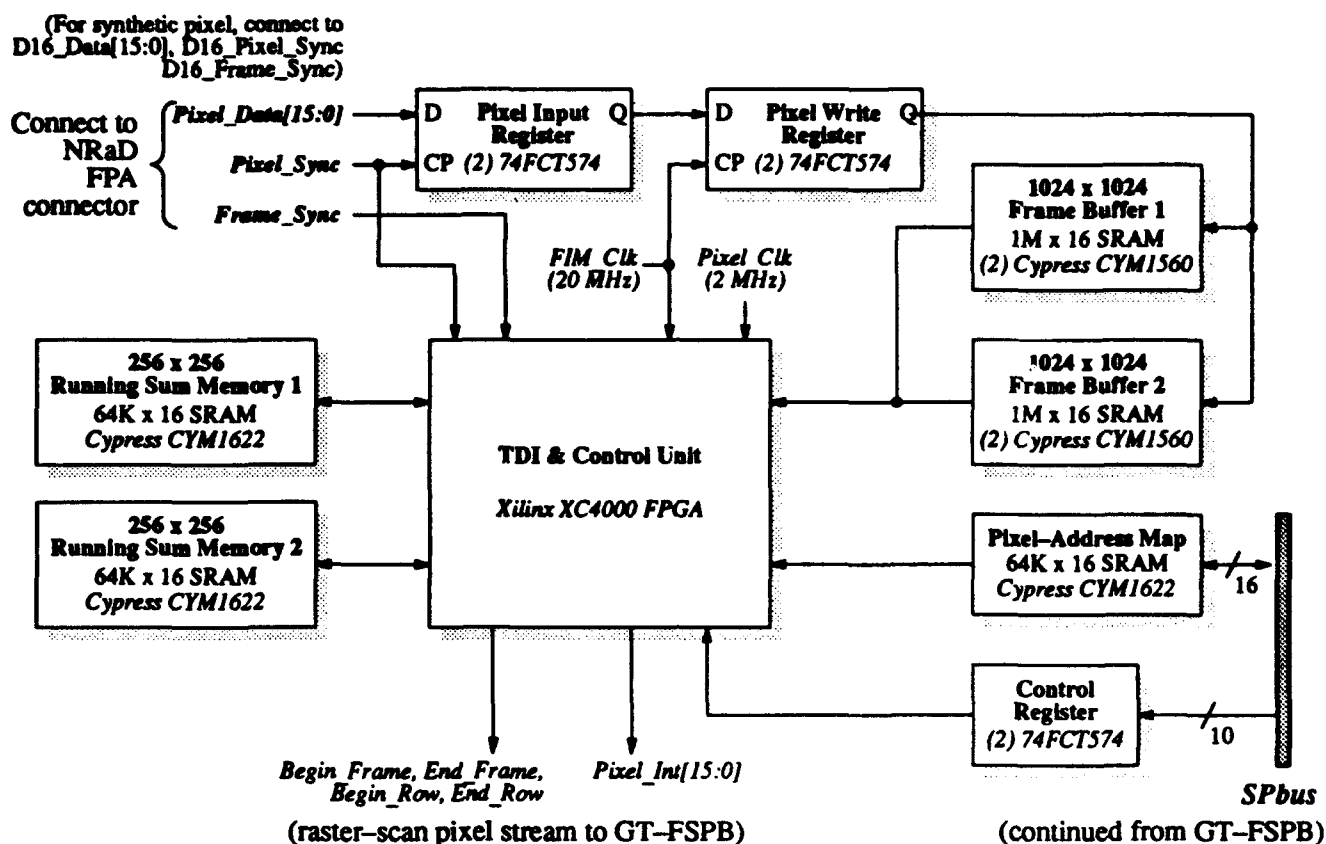


Figure 4. Georgia Tech FPA Interface Module (GT-FIM)

Figure NO TAG shows the hardware circuit diagram of GT-FFE. It consists of 128, 16Kx4 SRAM chips (IDT71982), sixteen bidirectional data transceivers (IDT 74FCT245), nineteen DFFs/registers (IDT 74FCT574), two 8-bit registered transceivers (IDT 74FCT646), and EPLDs for the control logic. The buffering of SRAM data and address signals is necessary to reduce capacitive loading and excessive propagation delays.

### 3.2.2. Software Design

The GT-XSPI graphical user interface (GUI) design is shown in Figures 6. The GUI gives a user control on the mode and display selection of the FPA input and SP outputs. It allows the user to analyze, in *real time*, the characteristics of an FPA sensor and the effectiveness of each image filtering process with respect to the sensor. Some changes have been made since the last report was written. They provide a better user interface and conform to SUN's *xview* library routines. Additional improvements are possible during the course of implementing the interface. The firmware (SBus driver) between GT-XSPI and the GT FPA Test System hardware will be developed next.

Figure 6 shows the screen/cockpit layout. Up to five real-time Display Windows can be activated by the user. The user can choose the Display Window size at integer multiples of 128x128, from 256x256 to 640x640 and position it anywhere on the screen. Area and intensity centroids are displayed in "+" and "x" shapes, overlaying the image of the selected Display Window(s).

Figure 7 shows the main command menu. A click on each button except "Halt" and "Quit" will bring up its corresponding control panel as shown in Figure 8 and 9. The size of each real-time Display Window can be resized to either 256x256, 384x384, 512x512 or 640x640. There are five Display Windows (FPA, Non-Uniformity Compensation, Temporal Filter, Spatial Filter and Thresholding) and each of them can be resized independently. A resizing request is made using the Main Option Panel in the Main Control Panel (Figure 8).

Besides window resizing, other standard windowing features such as zoom, close, and pan on a zoomed image are provided. There are two ways of zooming. One is menu driven zooming or "incremental zooming", and the other is mouse controlled zooming or "arbitrary zooming". The incremental zooming lets the user zoom in/out by selecting an appropriate button in the zoom menu of the display window. The arbitrary zooming lets the user zoom in on a specific pixel of interest quickly. As the user presses the middle button of the mouse and drags the mouse, a rectangular box appears. When the user releases the button, the zooming operation is initiated and the Display Window displays the pixels inside the bounding box. This operation can be enabled/disabled

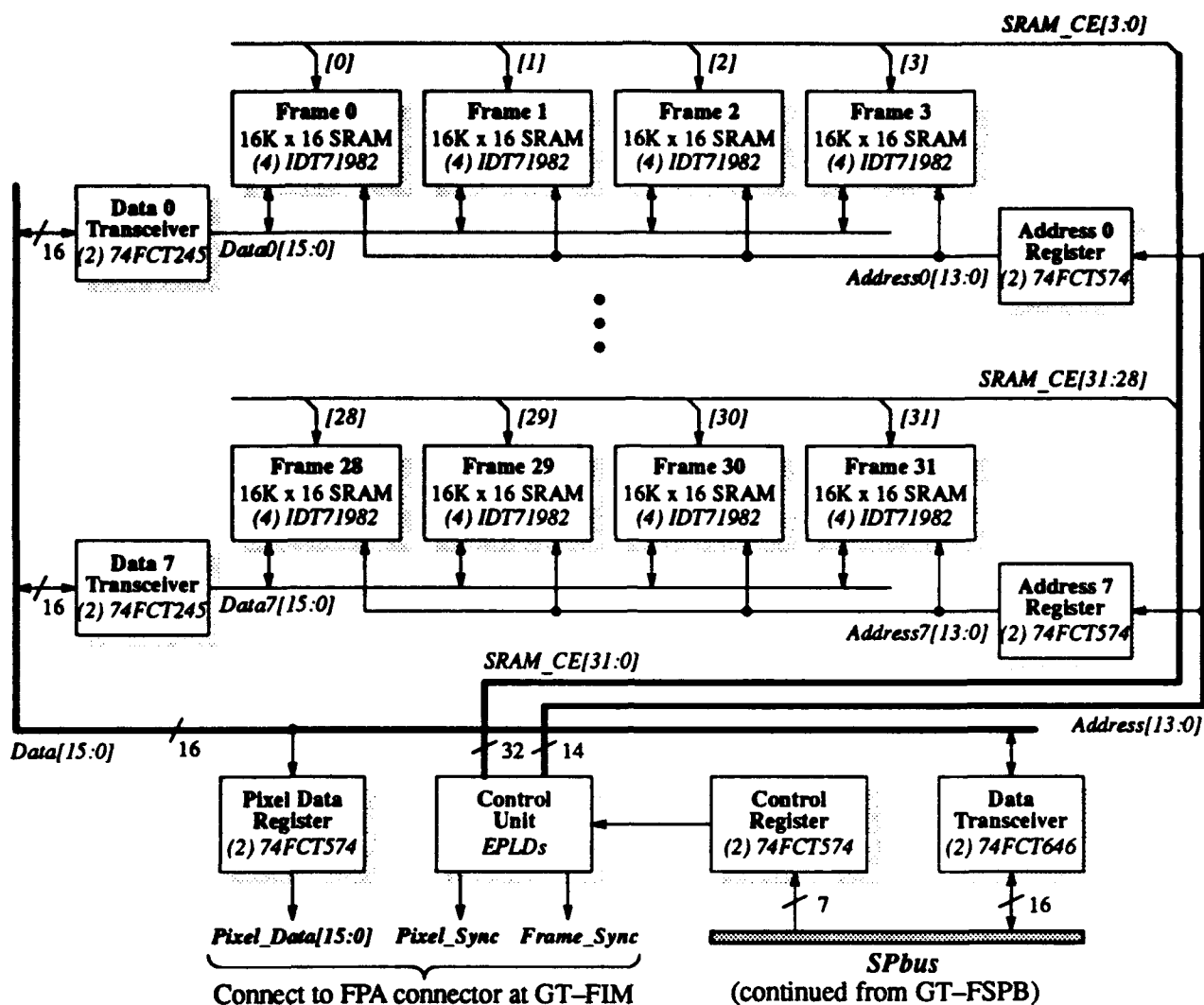


Figure 5. FPA Frame Emulator (GT-FFE)

through the Option Control Panel (Figure 9). There are two ways of panning. The user can pan the zoomed image either using arrow keys on a keyboard or mouse clicking on the pan window. By pressing the left-arrow key, the region being displayed in the Display Window shifts one FPA pixel to the left, by pressing the up-arrow key, it shifts one FPA pixel up, and so on. By clicking the mouse in the pan window, the displayed region shifts to where the mouse points to. By combining these two panning operations, the user can pan to the pixel of interest easily and precisely. Finally, if a user wants to monitor the pixel intensity while moving the cursor, the Pixel Value Window can be activated.

The GUI features provided allow a user to visually locate bad FPA detectors. The various Display Windows of filtered FPA images also allow a user to characterize the strengths and weaknesses of a particular FPA sensor with respect to color band, types of noise, background, etc. A user can visually compare the effects of non-uniformity compensation, temporal filtering, spatial filtering, and thresholding. The Program button allows a user to program the SP filter chips (NUC, TF, SF, and THR). The Program Panel example for the Spatial Filtering chip is shown in Figure 9. If certain effects are not desirable, the user can re-program the filter chips. For example, a user can experiment with different thresholding modes (simple, adjusted, adaptive) in the Thresholding chip. Note that the FPA Program Panel initializes parameters specific to the FPA sensor being tested, such as gain, offset, frame segment size (128x128 or 256x256), selected frame segment, pixel-address mapping (how the pixel data streams out), TDI factor (divide by 1, 2, 4, 8, or 16), and input source (real FPA sensor or host's synthetic FPA input image).

The Centroiding Control Panel has a different appearance from other control panels associated with filter chips, since it does not possess its own image display (Figure 10). The panel enables/disables overlay display of both area and intensity centroids on each display image separately.

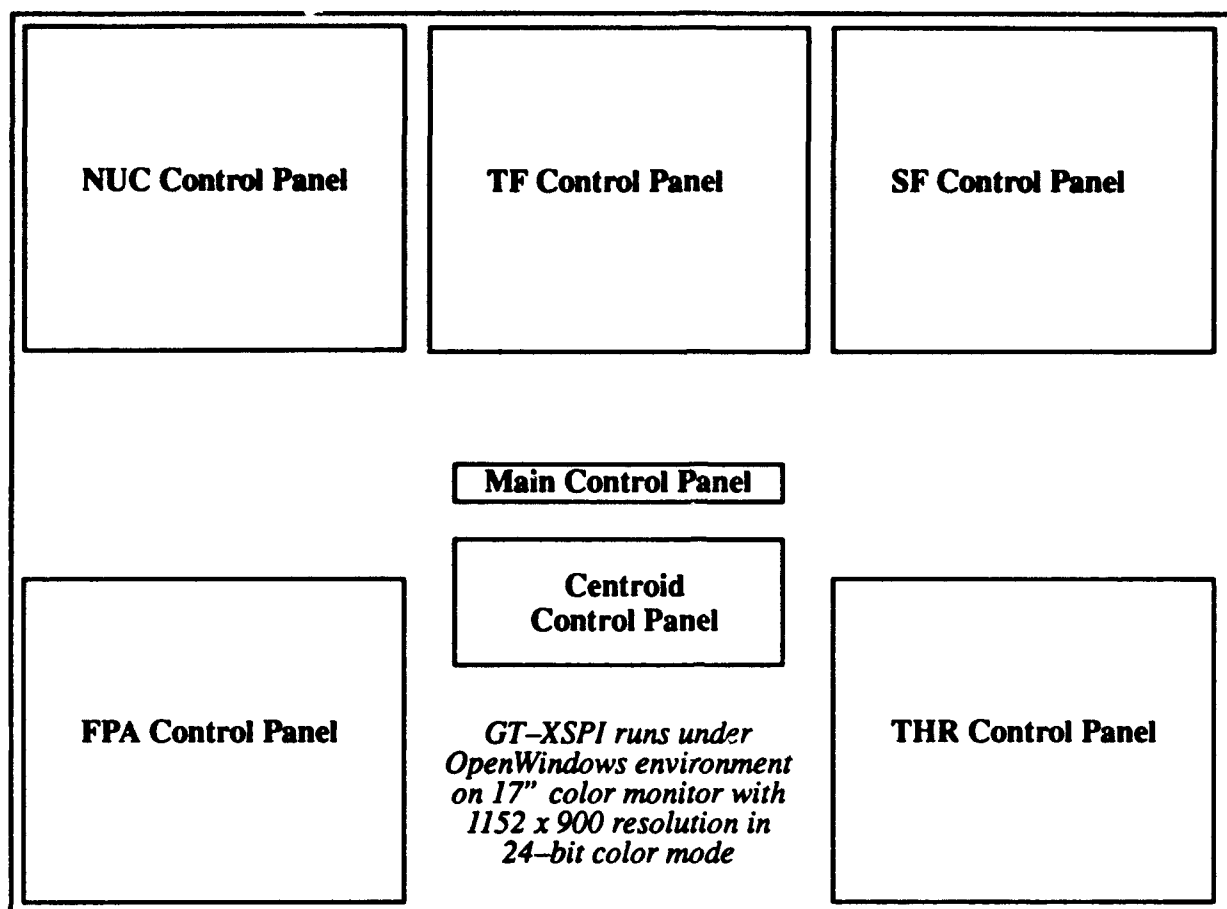


Figure 6. GT-XSPI Graphical User Interface



**Description:**

FPA Button pops up FPA Control Panel (controls raw Focal Plane Array modes & display)  
 NUC Button pops up NUC Control Panel (controls Non-Uniformity Compensation chip & display)  
 TF Button pops up TF Control Panel (controls Temporal Filtering chip & display)  
 SF Button pops up SF Control Panel (controls Spatial Filtering chip & display)  
 THR Button pops up THR Control Panel (controls Thresholding chip & display)  
 CNT Button pops up Centroid Control Panel (controls Centroiding display)  
 Option Button pops up Main Option Panel (controls various options of the interface)  
 Halt Button stops updating all the display windows.  
 Quit Button terminates the program.

Figure 7. GT-XSPI Main Control Panel

### 3.3. FPA Interface Specifications

Georgia Tech contacted Mr. Steve Frisbie at NRaD to clarify few things related to the interface between the NRaD FPA data acquisition system and the Georgia Tech FPA Test System. The following is a summary of the FPA interface specifications based on our conversation with Mr. Steve Frisbie on April 19, 1994.

Main Option Panel				
FPA	<input type="radio"/> 258x258	<input type="radio"/> 384x384	<input type="radio"/> 512x512	<input type="radio"/> 640x640
NUC	<input type="radio"/> 258x258	<input type="radio"/> 384x384	<input type="radio"/> 512x512	<input type="radio"/> 640x640
TF	<input type="radio"/> 258x258	<input type="radio"/> 384x384	<input type="radio"/> 512x512	<input type="radio"/> 640x640
SF	<input type="radio"/> 258x258	<input type="radio"/> 384x384	<input type="radio"/> 512x512	<input type="radio"/> 640x640
THR	<input type="radio"/> 258x258	<input type="radio"/> 384x384	<input type="radio"/> 512x512	<input type="radio"/> 640x640

Figure 8. Main Option Panel: Size Option for each Display Window

NRaD currently uses a 40-pin ribbon cable for the FPA digital interface signals. The interface signals are Frame\_Sync, Pixel\_Sync, and Pixel\_Data[15:0]. The Pixel\_Data[15:0] is a 16-bit 2's complement pixel intensity, which is the output of a 12-bit A/D converter and sign-extended to 16 bits. The rising edge of Pixel\_Sync indicates when Pixel\_Data can be latched in (see Figure 11). It is assumed that Pixel\_Data is stable 20 ns prior to the rising edge of Pixel\_Sync, and no longer driven afterward. Signal values might hold longer than 0 ns due to the "relatively slow" discharge of the load capacitance. The skew problem between the FPA Data Acquisition System (cable input) and the FPA connector at the GT-FIM (cable output) should not present a problem if the skews/delays at the cable output side of Pixel\_Sync and Pixel\_Data are relatively even. Note that a 12-ns skew is possible if a 6-foot cable is used, assuming a 2-ns propagation delay per foot of cable.

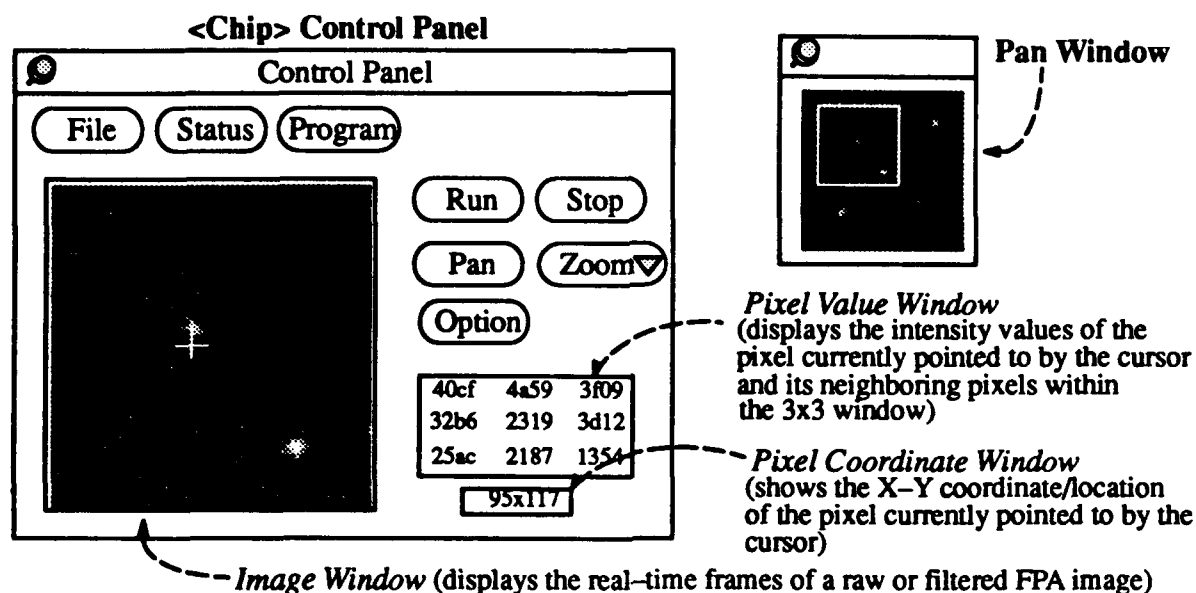
Pixels are not sent at a fixed rate, but in a "burst and idle" manner. The maximum burst speed from NRaD's FPA electronics is yet to be confirmed, but the GT-FIM is designed to support a burst speed of 20 MHz (50-ns minimum burst period) which is more than sufficient. The pixels are also not sent in a raster-scan order. Different FPA vendors have different pixel output sequences. To accommodate this, the GT-FIM is designed to handle total random ordering, by re-mapping *each* pixel-output-sequence number with its actual pixel address/location (user programmable by the GT-FIM Pixel-Address Map memory).

The pin description at the 40-pin FPA connector is as follows: (to be confirmed)

GND: all odd pin numbers 1, 3, 5, ..., 39  
Pixel\_Data0 ... Pixel\_Data15: even pin numbers 0, 2, 4, ..., 30  
Pixel\_Sync: pin 32  
Frame\_Sync: pin 34  
unused pins: 36, 38, 40.

The external cable used is a 100-Ohm ribbon cable (planned to be replaced by a twisted pair cable) approximately 6 feet long. The signal pins must be terminated at the receiving end (at GT-FIM). The line termination method is being investigated.

The Pixel\_Data is not always the true pixel intensity from the FPA sensor, since it can be adjusted by the user with a gain



**Description:**

File Button pops up File Control Panel  
 Program Button pops up Programming Panel  
 Run Button starts updating Display Window  
 Zoom Button pulls down Zoom Menu  
 Pan Button pops up Pan Window

Status Button pops up Status Panel

Stop Button stops updating the Display Window  
 Size Button pulls down Window Size Menu  
 Option Button pops up Option Control Panel

**Program Panel (example, of SF)**

**SF Program Panel**

Filter Coefficients

a	0.25
b	0.25
c	0.0
d	0.50

Reset

Apply

Quit

**File Control Panel**

Image

File Name \_\_\_\_\_ Save

Status

File Name \_\_\_\_\_ Save

**Option Control Panel**

Pixel Value Update ☐ On ☐ Off

Coordinate Update ☐ On ☐ Off

Arbitrary Zooming ☐ On ☐ Off

Grid ☐ On ☐ Off

**Zoom Menu**

256x256

384x384

512x512

640x640

768x768

Larger

Smaller

**Figure 9. Control Panel for FPA, NUC, TF, SF and THR Display Windows**

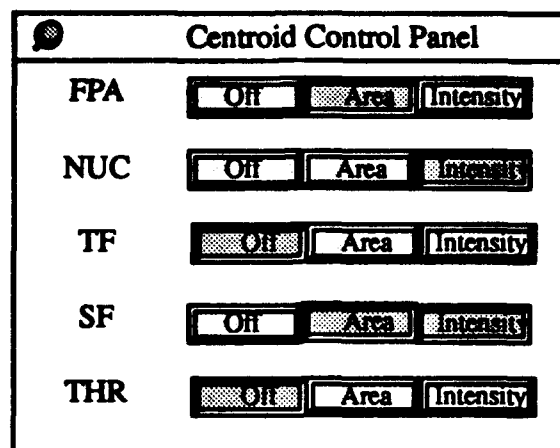


Figure 10. Command Submenus: Centroiding Control Panel

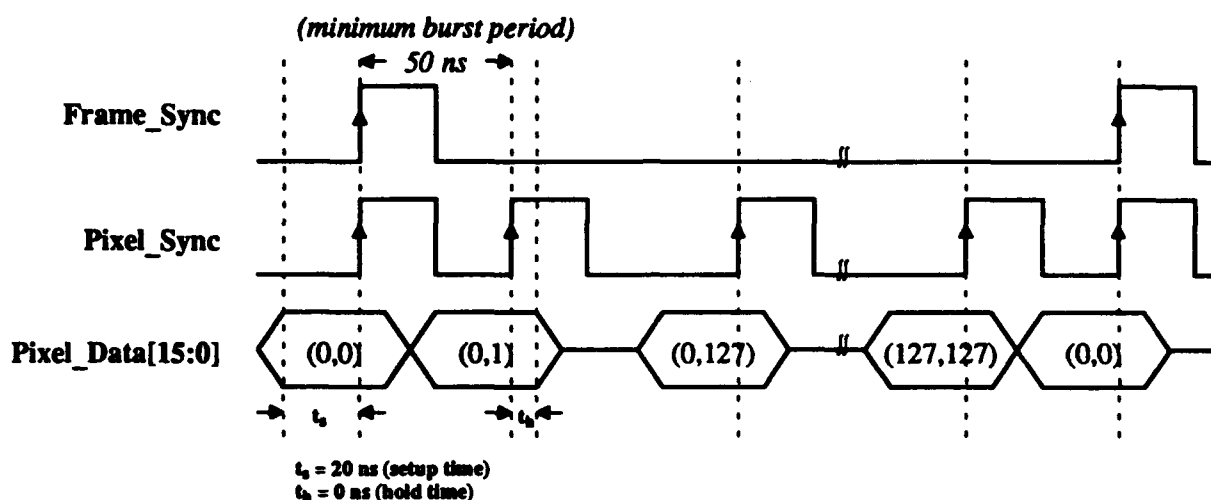


Figure 11. FPA Interface Signal Timing Diagram

and offset in the FPA data acquisition system:

$$\text{Pixel\_Data} = (\text{FPA\_pixel\_int} - \text{Offset}) * \text{Gain} . \quad (1)$$

This adjustment must be "reversed/corrected" at the GT FPA Test System to get the true pixel intensity, which can be done by the Non-Uniformity Compensation chip (GT-VNUC) as described in next section.

### 3.4. Gain & Offset Correction

#### 3.4.1. NRaD Test System

The laboratory data acquisition system at NRaD is constructed with the dataflow shown in Figure 12. The FPA under test is irradiated with some intensity. A single pixel will be subject to intensity  $I_i$ , and will produce an output  $O_i$ . This output can then be added to an Offset voltage,  $OS$ , and multiplied by a Gain,  $K$ . The analog value is then converted to a two's complement, 16-bit integer and sent to the Georgia Tech FPA Test System. The Offset and Gain are used to expand the output scale to cover the full range of the A/D converter. Two typical linearized responses are shown in Figure 11. Curve A represents the FPA output,  $O_i$ . Curve B represents the output,  $O_{ia}$ , after offset and gain have been applied. This section explains the way these signals will be processed to correct for gain and offset.

#### 3.4.2. Georgia Tech FPA Test System

The Georgia Tech FPA Test System includes a Non-Uniformity Compensation (NUC) chip which can model each pixel response as a set of four line segments, which are continuous and monotonically increasing. It should also be noted that the chip can model each pixel using 1, 2, or 3 line segments if the user chooses to do so. The following discussion assumes four line segments are used, but applies to all cases. The line segments are determined by using five known intensities,  $I_0, I_1, I_2, I_3$ , and  $I_4$ , and reading the pixel responses,  $O_0, O_1, O_2, O_3$ , and  $O_4$ , for each input. The input-output pairs are stored in the memory which

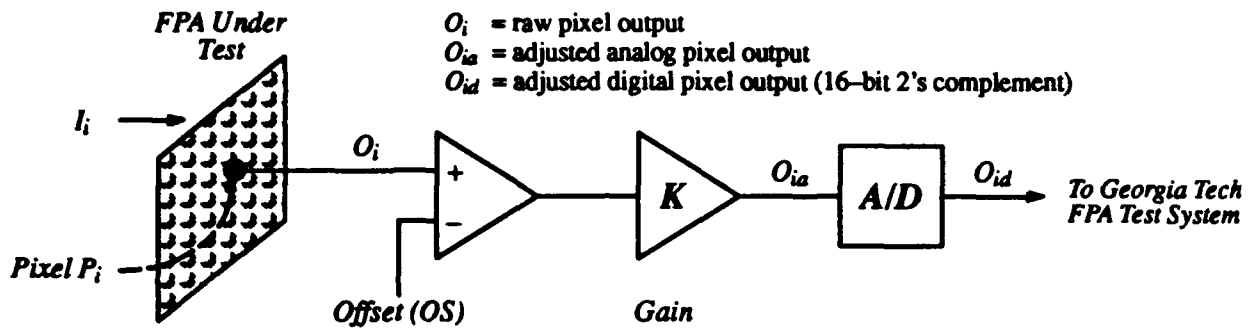


Figure 12. NRaD Laboratory Test System

supports the NUC chip. When an unknown FPA pixel output,  $O_x$ , is applied to the NUC chip, the output of the chip is the input pixel intensity as specified by one of the line segments. The chip first determines the line segment by searching the list of outputs until,  $O_i \leq O_x \leq O_{i+1}$ . The intensity value is then determined from the equation,

$$I_x = \frac{(O_x - O_i)(I_{i+1} - I_i)}{(O_{i+1} - O_i)} + I_i \quad (2)$$

where  $i = 0, 1, 2$ , or  $3$ .

Since the signed value,  $O_{id}$ , contains  $OS$  and  $K$ , the calibration data will automatically include these values as part of the line segment definitions. Thus, the NUC chip, when calibrated with the offset and gain inputs set to their proper values, will remove the gain and offset and produce the correct input as represented by an approximation of line segments.

### 3.4.3. Special Cases

There are two conditions that produce special cases. If the input,  $I_i$ , exceeds some level, the output,  $O_{ia}$  will saturate. However, the A/D output may saturate prior to this level. Hence there is some maximum,  $O_{max}$ , that limits the output for  $I_i > I_{max}$ . As shown in Figure 13 neither A nor B has saturated from the input  $I_i$ , but the output is limited by the A/D. Thus  $O_{ia}$  and  $O_{id}$  are not equal for the case shown in Figure 13 when the input exceeds the value which produces  $O_{max}$ .

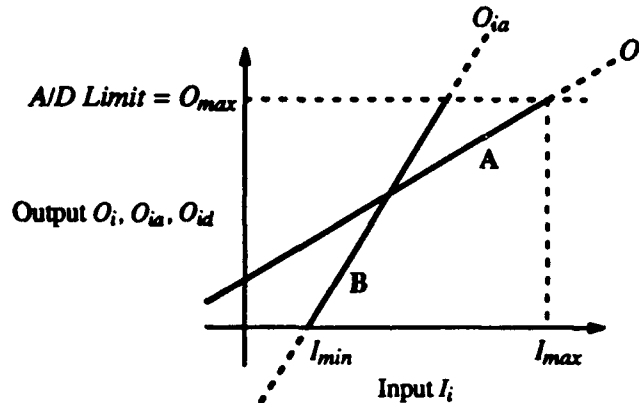


Figure 13. Linearized Pixel Responses

A second condition exists when the input is negative as shown in curve B of Figure 13. For  $I_i$  less than some value,  $I_{min}$ , the output  $O_{ia}$  will be negative. This value will be converted into an equivalent negative digital value,  $O_{id}$ . However, the NUC chip does not recognize negative values as legal inputs and will convert these values to zero. This raises the question of how these values are to be treated.

### 3.4.4. Design Methodology

The current plan is to input all values  $O_{id}$  to the NUC chip and use the chip to remove the gain and offset applied by the data acquisition system. This implies that all negative values will be set to zero, and all values greater than  $O_{max}$  will be set to  $O_{max}$ . This also implies that the NUC chip has to be calibrated with the appropriate Offset and Gain parameters set to the values that will be used during the test. If the calibration is carried out for one set of Offset and Gain values and a new setting is used during the test, it will be necessary to correct all the values in the NUC chip memory. This can be done but it is not the preferred



mode of operation. In all cases, the value  $O_M$  is made available for display and can be examined as a complete frame or a single pixel. However, this value of the output does have the data acquisition system offset and gain embedded in the value.